J Control Theory Appl 2013 11 (2) 237–246
DOI 10.1007/s11768-013-1186-z

Adaptive sampling immune algorithm solving joint chance-constrained programming

Zhuhong ZHANG†, Lei WANG, Min LIAO

Institute of System Science & Information Technology, College of Science, Guizhou University, Guiyang Guizhou 550025, China

Abstract: This work investigates one immune optimization algorithm in uncertain environments, solving linear or nonlinear joint chance-constrained programming with a general distribution of the random vector. In this algorithm, an *a priori* lower bound estimate is developed to deal with one joint chance constraint, while the scheme of adaptive sampling is designed to make empirically better antibodies in the current population acquire larger sample sizes in terms of our sample-allocation rule. Relying upon several simplified immune metaphors in the immune system, we design two immune operators of dynamic proliferation and adaptive mutation. The first picks up those diverse antibodies to achieve proliferation according to a dynamical suppression radius index, which can ensure empirically potential antibodies more clones, and reduce noisy influence to the optimized quality, and the second is a module of genetic diversity, which exploits those valuable regions and finds those diverse and excellent antibodies. Theoretically, the proposed approach is demonstrated to be convergent. Experimentally, the statistical results show that the approach can obtain satisfactory performances including the optimized quality, noisy suppression and efficiency.

Keywords: Joint chance-constrained programming; Immune optimization; Adaptive sampling; Reliability dominance; Noisy attenuation

1 Introduction

Many real-world engineering optimization problems, e.g., control system design, energy production and management [1], appear in uncertain environments; namely, they involve noisy factors. Such kind of problem, so-called stochastic optimization, is one challenging kind of optimization in the context of optimization, as the solution quality is influenced seriously by noise. Chance-constrained programming (CCP), originally proposed by Charnes and Cooper [2], is a special type of stochastic optimization, being composed of the expected value/deterministic objective function and chance constraint(s). The major difficulty of solving CCP includes two aspects. First, it is almost impossible to check directly whether a given candidate solution is feasible. Second, the feasible region is nonconvex in general [3]. Although some researchers paid great attention to such a hot topic, few achievements have been reported in the literature. Some early work mainly studied how to handle chance constraints under certain assumptions, for example, convexity approximation [4–7], logarithmically concave or transformation [2]. These theoretical results are greatly restricted in practical applications, due to their computational complexity or sophisticated transformation. Therefore, advanced optimization techniques are desired for CCP and especially for nonlinear CCP.

Immune optimization is a well-known hot branch in artificial immune systems [8], owing to great superiorities over several classical intelligent approaches when solving multimodal optimization problems. Although a series of excellent optimization approaches based on bio-immune inspirations, suitable for static or dynamic optimization problems, appear in the literature [9–12], these works are rarely done to study immune optimization for CCP problems. Since such kind of problem involves many important applications, we in this paper investigate an adaptive sampling immune algorithm (ASIA) for joint CCP (JCCP) with a general distribution of the random vector.

2 Related work on sampling and intelligent approaches

2.1 Sampling approaches

Chance constraints are probabilistic inequalities basically. Integral calculation and Monte Carlo approximation (MCA) are two conventional ways in dealing with such constraints. The first is usually adopted difficultly, due to some inherent characteristics of the integral formula, such as nonlinearity, discontinuation or multidimensionality, but the second is a simple and available method from the perspective of numerical simulation. Despite of a wide application, MCA arises a crucial issue, i.e., how to decide the sample size of the random vector at an individual (we simply say the sample size of the individual). Many researchers studied CCP problems in the precondition of static sampling strategies [2–5, 13–18]. For instance, Shapiro [13] analyzed the limitation behavior about the sample approximation of the true JCCP problem, and accordingly obtained a sample average approximation (SAA) method. Thereafter, Shapiro et al. [3–5, 15] claimed that the empirical minimum acquired by SAA could approach the theoretical optimal value when the sample size was sufficiently large. In addition, several researchers made great efforts to probe into

Received 15 September 2011; revised 16 August 2012.
†Corresponding author. E-mail: sci.zhzhang@gzu.edu.cn. Tel.: +86-13765033981; fax: +86-851-3620186.
This work was supported by the National Natural Science Foundation of China (No. 61065010), and the Doctoral Fund of Ministry of Education of China (No. 20125201110003).

a priori lower bound sample estimates [14, 19]. For example, Luedtke and Ahmed [14] explored the relationship of approximation between the sample approximation problem and the true JCCP. This investigation gives us a valuable guidance about how to decide the sample size of individual.

Adaptive sampling is an important method in reducing computational time. Higle et al. [20] made some empirically comparative studies for adaptive and static sampling, by means of two approaches of stochastic decomposition (SD) and SAA mentioned. The experimental results indicate that, with respect to the optimized quality, there is slight difference between SD and SAA, but SAA causes high computational complexity. Loughlin et al. [21] acquired a Latin hypercube sampling (LHS) approach suitable for chance constraints. Such approach, together with one genetic algorithm, can almost lead to the same solution quality as an excellent evolutionary algorithm with the fixed sample size, while only requiring low computational cost. In addition, simulation-based optimization is an important sub-area, in which the main work is to develop a sample-allocation scheme that maximizes the probability of correct selection. Such kind of scheme does not involve any constraints, in particular chance constraints. For instance, Chen and Lin [22] developed an efficient dynamic sample allocation scheme to decide the best of individuals in a given population. Subsequently, Chen and He [23] proposed an extended version of such scheme which was used to find the top-m individuals among k individuals, only relying upon observed objective values and variances of individuals. The experiments indicate that such two schemes are excellent. However, through the process of their designs, we notice that, although these two schemes are capable of solving expected value optimization problems without any constraint, they do not suit CCP problems, due to chance constraints. Furthermore, depending on one kernel density function, Sahin and Diwekar [24] developed a mathematical optimization approach to the kind of CCP with probabilistic objective functions, in which the computational burden of determining the search direction was reduced in terms of a re-weighting method. In our previous work [25–26], we developed two kinds of adaptive sampling schemes. One decides the sample size of individual according to whether an individual is empirically feasible [25], and the other [26], based on the hypothesis test, is designed to emphasize that better individuals get larger sample sizes. Note that the optimizer in [26] is effective only for nonconstrained expected value optimization problems, instead of CCP problems.

2.2 Intelligent approaches handling CCP

Although evolutionary algorithms are a popular optimizer for CCP [3, 16–18, 21, 27], they are applied mainly to linear stochastic programming problems which can be equivalently formulated by analytically deterministic models. For example, Tanner [27] in his doctoral thesis investigated in detail the linear JCCP problem with the random left-hand side, by reformulating it into an analytically equivalent mixed-integer programming. In his work, a Tabu search meta-heuristic procedure was provided when finding good feasible solutions. Tanner and Ntaimo [28] also studied optimality cuts called IIS (irreducibly infeasible subsets) cuts, and hence, developed a branch-and-cut algorithm based on IIS cuts. In addition, the recent research on stochastic optimization has exposed that some researchers keep their eyes on nonlinear stochastic programming. Brand [19] explored the possibility of the reformulation for nonlinear JCCP problems by suitably selecting penalty-type objectives. His theoretical result shows that the transformed model can asymptotically approach the true one.

A few researchers made an encouraging attempt to integrate stochastic simulation with intelligent optimization techniques in order to deal effectively with CCP [16–18]. For example, Liu and Xiao [16–17] presented their respective optimization approaches related to the BP neural network. The main difference between them consists in their different evolutionary fashions; namely, Liu found the desired solution by using one conventional genetic algorithm, but Xiao did so in terms of one particle swarm optimization appraoch. Poojari et al. [18] investigated nonjoint CCP problems by developing two types of genetic algorithms with subtle differences, respectively simply written as SSGA-I and SSGA-II. Their main difference involves different optimality scoring rules as associated to fitness evaluations. Özcan [29] developed a real-world industrial production program model described by a chance-constrained, piecewise-linear, mixed integer optimization one, and hence, solved it by means of the simulated annealing algorithm.

Although immune optimization is a hot and potential topic in the area of artificial immune systems, it is rarely studied for CCP problems. Recently, we developed a simple immune algorithm (i.e., IOM) to handle nonjoint CCP [25]. In such work, one of concerns is to investigate a sampling scheme which guarantees that all empirically feasible individuals in the current population share the same larger sample size than empirically infeasible ones. However, this sampling scheme needs to make further improvements, e.g., different individuals should obtain different sample sizes. In addition, Qu [30] set up a CCP model which formulated the optional selection problem of investment projects. Then, an immune clonal selection algorithm with the fixed sample size was designed to provide a general solution for the model.

In our present work, we first suggest an *a priori* bound estimate to handle chance constraints with a general distribution of the random vector. Second, an adaptive sampling scheme, based on the concept of reliability-dominance, is designed to allocate the population sample size to different individuals, and then it is integrated with an immune evolution mechanism to form ASIA as mentioned. Here, it is emphasized that ASIA is different from our previous IOM [25] and other reported work. Although ASIA and IOM share some common immune characteristics and assume solving CCP problems, they are different optimizers with major differences. First, their tasks are different. ASIA handles JCCP, but IOM is effective for nonjoint CCP problems. ASIA processes the chance constraint by an *a priori* estimate, and requires that the reliability-dominance based adaptive sampling scheme make different individuals get different sample sizes. This does not present in IOM. Second, their immune inspirations cause different design ideas.

3 Problem description

Consider the following JCCP problem of form (P_α):

$$\min_{\boldsymbol{x}\in D} \mathrm{E}[f(\boldsymbol{x},\xi)]$$
$$\text{s.t.} \begin{cases} \Pr\{\boldsymbol{G}(\boldsymbol{x},\xi)\leqslant \boldsymbol{0}\}\geqslant 1-\alpha, \\ \boldsymbol{g}(\boldsymbol{x})\leqslant \boldsymbol{0}, \boldsymbol{h}(\boldsymbol{x})=\boldsymbol{0}, \end{cases}$$

with bounded and closed domain D in $\mathbb{R}^p$, decision vector $\boldsymbol{x}$ in D and significance level α in the interval $(0,1)$, where ξ is a general q-dimensional real valued random vector with unknown prior distribution information; $E[\cdot]$ and $\Pr\{\cdot\}$ are the operators of expectation and probability, respectively; $f(\cdot,\cdot)$ denotes the linear or nonlinear stochastic objective function; $\boldsymbol{G}(\boldsymbol{x},\xi)$ is the linear or nonlinear stochastic vector-valued constraint function taking values in $\mathbb{R}^I$ with $\boldsymbol{G}(\boldsymbol{x},\xi)=[G_1(\boldsymbol{x},\xi)\ \cdots\ G_I(\boldsymbol{x},\xi)]^{\mathrm{T}}$; $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ are the deterministic vector-valued constraint functions with $\boldsymbol{g}(\boldsymbol{x})=[g_1(\boldsymbol{x})\ \cdots\ g_J(\boldsymbol{x})]^{\mathrm{T}}$ and $\boldsymbol{h}(\boldsymbol{x})=[h_1(\boldsymbol{x})\ \cdots\ h_K(\boldsymbol{x})]^{\mathrm{T}}$. Here, we prescribe that the symbol of $\boldsymbol{x}\leqslant\boldsymbol{0}$ stands for $x_i\leqslant 0$ with $\boldsymbol{x}=(x_1,x_2,\ldots,x_p)$ and $1\leqslant i\leqslant p$. We say that $\boldsymbol{x}$ is reliable with the significance level α, or say that $\boldsymbol{x}$ is a reliable candidate solution, if it satisfies the above constraints; otherwise, it is called unreliable. A reliable candidate solution is called an optimal reliable solution if it possesses the minimal objective value among reliable candidate solutions. All such optimal reliable solutions consist of set O_α^*. Here, a constraint violation function is introduced,

$$\begin{aligned}\Gamma(\boldsymbol{x})=&\max\{1-\alpha-p(\boldsymbol{x}),0\}\\&+\sum_{j=1}^{I}\max\{g_j(\boldsymbol{x}),0\}+\sum_{k=1}^{K}|h_k(\boldsymbol{x})|,\end{aligned}\tag{1}$$

where $p(\boldsymbol{x})=\Pr\{\boldsymbol{G}(\boldsymbol{x},\xi)\leqslant\boldsymbol{0}\}$. Obviously, $\boldsymbol{x}$ is reliable if $\Gamma(\boldsymbol{x})=0$. Additionally, if P_α includes multiple joint chance constraints, they are similarly embedded to such equation. Based on equation (1), we introduce the concept of reliability-dominance to compare two candidate solutions.

Definition 1 (Reliability-dominance) [23] Let $\boldsymbol{x},\boldsymbol{y}\in D$, we say that $\boldsymbol{x}$ dominates $\boldsymbol{y}$ with a given significance level (simply write $\boldsymbol{x}\prec\boldsymbol{y}$), if one of the following conditions holds:

a) $\boldsymbol{x}$ and $\boldsymbol{y}$ are reliable, and $\mathrm{E}[f(\boldsymbol{x},\xi)]<\mathrm{E}[f(\boldsymbol{y},\xi)]$;

b) $\boldsymbol{x}$ is reliable, but $\boldsymbol{y}$ is not;

c) $\boldsymbol{x}$ and $\boldsymbol{y}$ are not reliable, but $\Gamma(\boldsymbol{x})<\Gamma(\boldsymbol{y})$.

In the following section, we will investigate a lower bound estimate of probability presented in the above chance constraint, so as to find an approximate optimal reliable solution. We also design a sample-allocation scheme to adjust the sample size of individual, which helps for reducing the magnitude of individual evaluation.

4 Approximation and adaptive sampling

4.1 Chance constraint approximation

Let $n(\boldsymbol{x})$ be the sample size of the q-dimensional real valued random vector ξ at the point $\boldsymbol{x}$, and $\xi_1,\xi_2,\ldots,\xi_{n(\boldsymbol{x})}$ be i.i.d. random vectors. Write $z_i=I(G(\boldsymbol{x},\xi_i)\leqslant\boldsymbol{0})$, where $I(\cdot)$ is an indicator function taking 1 if '$\cdot$' is true and 0 otherwise. Thus, $z_1,z_2,\ldots,$ and $z_{n(\boldsymbol{x})}$ follow the binomial distribution $B(1,p(\boldsymbol{x}))$. Furthermore, through the central limit theorem, we know that the sample average approximation $p_{n(\boldsymbol{x})}$ asymptotically follows the normal distribution $N(p(\boldsymbol{x}),p(\boldsymbol{x})(1-p(\boldsymbol{x}))/n(\boldsymbol{x}))$ [31], where $p_{n(\boldsymbol{x})}=n(\boldsymbol{x})^{-1}\sum_{i=1}^{n(\boldsymbol{x})}z_i$. Take a random variable u,

$$u=\frac{p_{n(\boldsymbol{x})}-p(\boldsymbol{x})}{\sqrt{\dfrac{p(\boldsymbol{x})(1-p(\boldsymbol{x}))}{n(\boldsymbol{x})}}}.\tag{2}$$

Accordingly, u follows asymptotically the standard normal distribution. Furthermore, for the above significance level α, it is easy to obtain that

$$\Pr\{|u|\leqslant u_{1-\frac{\alpha}{2}}\}=2\Phi(u_{1-\frac{\alpha}{2}})-1=1-\alpha,\tag{3}$$

where $\Phi(\cdot)$ is the cumulative distribution function. We take

$$|p_{n(\boldsymbol{x})}-p(\boldsymbol{x})|\leqslant u_{1-\frac{\alpha}{2}}\sqrt{\frac{p(\boldsymbol{x})(1-p(\boldsymbol{x}))}{n(\boldsymbol{x})}},\tag{4}$$

which yields

$$(1+\frac{c}{n(\boldsymbol{x})})p(\boldsymbol{x})^2-(2p_{n(\boldsymbol{x})}+\frac{c}{n(\boldsymbol{x})})p(\boldsymbol{x})+p_{n(\boldsymbol{x})}^2\leqslant 0,\tag{5}$$

with $c=u_{1-\frac{\alpha}{2}}^2$. Therefore, equation (5) implies

$$\begin{aligned}p(\boldsymbol{x})\geqslant&(1+\frac{c}{n(\boldsymbol{x})})^{-1}[p_{n(\boldsymbol{x})}+\frac{c}{2}n(\boldsymbol{x})\\&-\sqrt{\frac{cp_{n(\boldsymbol{x})}(1-p_{n(\boldsymbol{x})})}{n(\boldsymbol{x})}+\frac{c^2}{4}n(\boldsymbol{x})^2}].\end{aligned}\tag{6}$$

Since $p_{n(\boldsymbol{x})}$ can asymptotically approach $p(\boldsymbol{x})$ when increasing $n(\boldsymbol{x})$, we take $\hat{p}_{n(\boldsymbol{x})}$ as a lower bound estimate of $p(\boldsymbol{x})$,

$$\hat{p}_{n(\boldsymbol{x})}=p_{n(\boldsymbol{x})}-u_{1-\frac{\alpha}{2}}\sqrt{\frac{p_{n(\boldsymbol{x})}(1-p_{n(\boldsymbol{x})})}{n(\boldsymbol{x})}}.\tag{7}$$

This way, in order to find the optimal reliable solution(s) to problem P_α, we only need to consider the following approximation problem (P_α^n)

$$\min_{\boldsymbol{x}\in D}\mu_{n(\boldsymbol{x})}(f)$$
$$\text{s.t. } \hat{p}_{n(\boldsymbol{x})}\geqslant 1-\alpha,\ \boldsymbol{g}(\boldsymbol{x})\leqslant\boldsymbol{0},\ \boldsymbol{h}(\boldsymbol{x})=\boldsymbol{0},$$

where $\mu_{n(\boldsymbol{x})}(f)$ denotes the the empirically average objective value at the point $\boldsymbol{x}$ attached sample size $n(\boldsymbol{x})$. Meanwhile, $\boldsymbol{x}$ is said to be an empirically reliable candidate solution, if it satisfies the above constraints. Notice that if $n(\boldsymbol{x})=M$ for all $\boldsymbol{x}\in D$, i.e., all candidate solutions are attached the same sample size M, P_α^n is a conventional approximation problem of P_α [15]. In this paper, we require that $n(\boldsymbol{x})$ depend on $\boldsymbol{x}$.

4.2 Adaptive sampling

Because of random factors, it is easy to deem inferior candidate solutions as superior ones during the solution process, and as a result the optimized quality is influenced seriously. Therefore, in order to acquire an approximate optimal reliable solution to the above JCCP problem, we in this paper require that the sample size of each candidate solution be determined dynamically. To this point, let X be a given population with size N. Assume that all empirical values for elements in X at the $(n-1)$th moment, i.e., empirically average values and empirical constraint violations related to equation (1), are known. Let M_n be the total of samples for

all elements in X at the nth moment given by

$$M_n = \mathrm{round}(m_0 N\sqrt{1+n}), \tag{8}$$

where $\mathrm{round}(v)$ is the maximal integer not beyond v; m_0 is a fixed integer taking 3 in this paper. Based on Definition 1, assume that $d(\boldsymbol{x})$ represents the number of individuals in X empirically dominated by $\boldsymbol{x}$ with $\boldsymbol{x} \in X$, and accordingly,

$$d(\boldsymbol{x}) = |\{\boldsymbol{y} | \boldsymbol{x} \prec \boldsymbol{y}, \boldsymbol{y} \in X\}|. \tag{9}$$

We easily know that $d(\boldsymbol{x}) > d(\boldsymbol{y})$, if $\boldsymbol{y}$ is empirically dominated by $\boldsymbol{x}$. Especially, if all elements in X are empirically reliable, the best candidate solution will get the largest dominance number. Based on such consideration, the sample size of candidate $\boldsymbol{x}$ in X is updated by

$$n(\boldsymbol{x}) = \mathrm{round}\Big(\frac{M_n d(\boldsymbol{x})}{\sum\limits_{\boldsymbol{y}\in X} d(\boldsymbol{y})}\Big). \tag{10}$$

Obviously, the total of sample sizes for elements in X is not beyond M_n; in addition, we know that $n(\boldsymbol{x}) > n(\boldsymbol{y})$ if $\boldsymbol{x} \prec \boldsymbol{y}$ with $\boldsymbol{x}, \boldsymbol{y} \in X$.

As related to the design of equation (10), we easily obtain the following property by means of the law of large numbers and equations (6) and (7).

Lemma 1 If $\boldsymbol{x} \in D$ is a reliable candidate solution to P_α satisfying $p(\boldsymbol{x}) > 1-\alpha$, there exists $N_0(\boldsymbol{x}) > 0$ such that, for $n > N_0(\boldsymbol{x})$, $\boldsymbol{x}$ is an empirically reliable candidate solution to P_α^n.

5 Immune theory and algorithm description

5.1 Clonal selection principle

The clonal selection principle essentially describes a learning process that B cells learn the invader and ultimately eliminate it. When an organism is exposed to the invader (antigen), a second signal from T_h cells stimulates the antigenic receptors of a B-cell to bind to such invader. This makes those high-affinity B cells create some matured plasma cells and memory ones through proliferation and somatic maturation. If such plasma cells are active, they secrete some antibodies neutralizing the triggering invader. In addition, the memory cells will become long-lived ones. Once the previous invader is found in the immune system, these memory cells commence rapidly differentiating into plasma cells capable of producing high-affinity antibodies. Such theory includes three main immune metaphors [32].

A) Cell selection. Those B cells with high affinities to the invader are chosen to change their pattern structures so that better B cells can be found.

B) Clonal expansion. Those stimulated B cells proliferate and differentiate into two different cell types. The plasma cells replicate their clones with their clonal sizes proportional to their affinities. The memory ones will live in the immune system for a long time.

C) Hypermutation. During the clonal expansion, genetic drift is introduced in the variable region. Occasionally, one such change leads to an increase in the affinity of the lymphocytes. This process creates a variety of new B cells, where the mutation probability of a B-cell is inversely proportional to its affinity to the antigen. After so, some worse clonal cells will encounter suppression.

By means of taking an analogy between the two processes of the immune response and solving JCCP, it is not difficult to know that some biological inspirations are useful for JCCP, because the task of such response is to create excellent B cells capable of eliminating the invader, and the purpose of handling JCCP is to find the optimal reliable solution. From the angle of engineering applications, the process of the above response may be simply simulated to construct our ASIA for JCCP.

5.2 Algorithm description

As associated with problem P_α^n in Section 4, a real-encoded reliable antibody is viewed as an empirically reliable candidate solution in the sense of the given significance level α; conversely, an empirically unreliable candidate solution is regarded as a unreliable antibody. The antigen is consistent with the problem itself. Our task is to find the best antibody (i.e., the optimal reliable solution) through running ASIA. In this paper, ASIA is composed of three main modules including adaptive sampling, dynamic proliferation and adaptive mutation. Given an antibody population X, each antibody's affinity in X at the nth moment is designed as follows:

$$\mathrm{aff}(\boldsymbol{x}) = \begin{cases} -\mu_{n(\boldsymbol{x})}(f), & \Gamma(\boldsymbol{x}) = 0, \\ \mathrm{aff}_{\min} - N(N - R(X))\Gamma(\boldsymbol{x}), & \text{otherwise}, \end{cases} \tag{11}$$

where $\Gamma(\boldsymbol{x})$ takes the empirical constraint violation of $\boldsymbol{x}$ with sample size $n(\boldsymbol{x})$, depending on equation (1); $\mathrm{aff}_{\min}$ and $R(X)$ are the minimal of affinities of all the reliable antibodies and the number of such antibodies in X, respectively. Especially, when all elements in X are unreliable, $\mathrm{aff}_{\min}$ takes 0. Equation (11) indicates that the affinity of each reliable antibody is larger than that of any unreliable one. After so, ASIA is described as follows:

Step 1 Set $n \leftarrow 1$. Generate an initial population A_n of N random antibodies, where each antibody is specified the same sample size m_0.

Step 2 Execute adaptive sampling on A_n, and evaluate all the elements in A_n by equation (11).

Step 3 Perform dynamic proliferation on A_n, and create a clonal population B_n.

Step 4 Carry out adaptive mutation on B_n, and obtain population C_n.

Step 5 Combine C_n and A_n, and select N antibodies with higher affinities to constitute the next population A_{n+1}.

Step 6 If the termination criterion is not satisfied, set $n \leftarrow n+1$ and return Step 2; otherwise, end the procedure.

In the above algorithm, each antibody acquires a reasonable sample size decided by its empirical reliability-dominance number through equation (10). Steps 2–5 are a loop of evolution which creates those excellent antibodies.

5.3 Immune operators

A) Dynamic proliferation. This mechanism first ranks decreasingly all elements in a given population X with size N, according to equation (11). Second, the clonal size of antibody $\boldsymbol{x}$ is decided by

$$l(\boldsymbol{x}) = |\{\boldsymbol{y} | 0 \leqslant \mathrm{aff}(\boldsymbol{x}) - \mathrm{aff}(\boldsymbol{y}) \leqslant r(\boldsymbol{x}), \boldsymbol{y} \in X\}|, \tag{12}$$

where $r(\boldsymbol{x})$ is a suppression radius index given by

$$r(\boldsymbol{x}) = \nu \sqrt{\frac{N}{1 + \sum\limits_{\boldsymbol{y}\in X} (\mathrm{aff}(\boldsymbol{x}) - \mathrm{aff}(\boldsymbol{y}))^2}}, \tag{13}$$

with an adjustable parameter ν in $(0, 1)$. Notice that an antibody cannot proliferate any clone if suppressed by some antibody, where we say that $\boldsymbol{y}$ is suppressed by $\boldsymbol{x}$ with $\boldsymbol{y} \neq \boldsymbol{x}$ if $\mathrm{aff}(\boldsymbol{y})$ is between $\mathrm{aff}(\boldsymbol{x}) - r(\boldsymbol{x})$ and $\mathrm{aff}(\boldsymbol{x})$. Therefore, all clones created by those survival antibodies constitute a clonal population with size N.

Equations (12) and (13) hint that antibody $\boldsymbol{x}$ suppress more antibodies if $l(\boldsymbol{x})$ and $r(\boldsymbol{x})$ are larger, and hence, proliferate more clones. This helps those survival antibodies maintain sufficient diversity.

B) Adaptive mutation. Let Y be a given clonal population, and X be the corresponding parent population. All clones in Y are mutated through the well-known polynomial mutation. Their mutation probabilities are conversely proportional to the affinities of their respective parents in X, given by

$$p_m(\boldsymbol{x}) = 1 - \beta \exp\{-(\frac{\gamma}{1+n} + w(\boldsymbol{x}))\}, \qquad (14)$$

with $0 < \beta < 1$ and $\gamma > 1$, where

$$w(\boldsymbol{x}) = \frac{\max\limits_{\boldsymbol{y}\in X}\{\mathrm{aff}(\boldsymbol{y})\} - \mathrm{aff}(\boldsymbol{x})}{\max\limits_{\boldsymbol{y}\in X}\{\mathrm{aff}(\boldsymbol{y})\} - \min\limits_{\boldsymbol{y}\in X}\{\mathrm{aff}(\boldsymbol{y})\}}. \qquad (15)$$

Thereafter, each of the clones mutated is allocated the same sample size m_n, and their empirical averages and constraint violations are also calculated.

Note that when executing theoretical analysis, we require that m_n take the sample size of the best antibody in the current population A_n. In practical applications, it takes a relative rational value, e.g., 30.

6 Computational complexity and convergence

In this section, some theoretical results on ASIA are derived, and their proofs are given in Appendix. Through the above description of algorithm and the designs of immune operators, ASIA's computational complexity is decided by Step 2. We obtain the following conclusion.

Theorem 1 The computational complexity in the worst case is $\mathrm{O}(m_0 N I\sqrt{n} + N(J+K) + N^2)$.

We next need to make the following assumption on P_α so as to investigate ASIA's convergence.

H1) O_α^* is nonempty and finite; $p(x^*) > 1 - \alpha$ for any $x^* \in O_\alpha^*$.

As we know, ASIA can be considered as an evolution chain: $A_n \to B_n \to C_n \to A_{n+1}$. Through the description of ASIA, A_{n+1} only depends on the state of A_n, while the mutation rate $p_m(\boldsymbol{x})$ as in equation (14) is dependent on n. Therefore, $\{A_n\}_{n\geqslant 1}$ is a nonhomogeneous Markov chain. Assume that the decision domain D as in Section 3 is a finite set, $D = \{\boldsymbol{x} = (x_1, x_2, \ldots, x_p) | x_i \in P_i, 1 \leqslant i \leqslant p\}$, where P_i is a set of r_i equal division points in the interval $[a_i, b_i]$. Let $\mathcal{S}$ represent the antibody space (i.e., reliable candidate solution space of P_α); $\mathcal{S}^N$ stands for a state space composed of antibody populations with sizes N, and $X \in \mathcal{S}^N$ is called a state with $X = (X_1, X_2, \ldots, X_N)$. $\Pr\{X \to Y\}$ denotes the probability which X is transformed into Y through the adaptive mutation above with $Y \in \mathcal{S}^N$. As associated to ASIA's formulation, we acquire the following properties.

Lemma 2 For $X, Y \in \mathcal{S}^N$, there exists δ, $0 < \delta < 1$, such that $\Pr\{X \to Y\} \geqslant \delta$.

Lemma 3 If the above hypothesis H1) holds, there exists $N_0 > 0$ such that when $n > N_0$, $\Pr\{A_{n+1} \cap O_\alpha^* = \varnothing | A_n \cap O_\alpha^* \neq \varnothing\} = 0$.

Lemma 4 For the same δ and N_0 above, $\Pr\{A_{n+1} \cap O_\alpha^* \neq \varnothing | A_n \cap O_\alpha^* = \varnothing\} \geqslant \delta$.

Theorem 2 If H1) holds, ASIA is convergent for any initial population distribution, i.e.,

$$\lim_{n\to\infty} \Pr\{A_n \cap O_\alpha^* \neq \varnothing\} = 1.$$

7 Numerical experiments

To examine ASIA's performance, four kinds of algorithms suitable for CCP, i.e., three representative approaches of HPSO [17], SSGA-I and SSGA-II [18], as well as our previous approach IOM [25], are chosen to compare with it, relying upon the following three test problems of Examples 1–3. Our experiments are executed on a personal computer with CPU/3 GHz and RMB/2 GB. Notice that SSGA-I and SSGA-II are two recent similar approaches as mentioned in Section 1. Such two approaches share the same fixed sample size for each individual, while their parameter settings, except the fixed population size taking 40 after experimental tuning, are the same as those in the literature [18]. In addition, HPSO is a hybrid method which integrates the BP neural network into a particle swarm optimization approach, where the network with 3000 training samples is utilized to evaluate individuals and to check whether an individual is empirically reliable. In such approach, we designate the population size as 30, and other parameter settings, except those of the network given by Table 1 (I: input layer, H: hidden layer, O: output layer, LR: learning rate, and LT: learning times), are the same as those in [17]. In IOM, the settings of parameters are taken in [25]. In ASIA, we specify the size of population as 40, and also take $\nu = 0.1, \beta = 0.98$ and $\gamma = 1.5$ after also experimental tuning. Furthermore, these five algorithms are required to execute respectively 30 single runs on each test problem, and their same termination criterion is that the total of evaluations during the solution process is 1.5×10^5. Furthermore, after solving all the test problems, we re-evaluate 3×10^5 times for the resultant solutions obtained by the algorithms, and the resultant empirical values are taken as the theoretical values of the solutions.

Table 1 Parameter settings of BP network.

Example	I	H	O	LR	LT
1	9	17	2	0.13	1
2	2	18	2	0.65	30
3	3	20	2	0.35	20

Example 1 (Open storage networks) [2]

$$\begin{aligned}
&\min \mathrm{E}[\eta + \sum_{i=1}^{9} |x_i|]\\
&\text{s.t.}\\
&\Pr\left\{\begin{array}{l} 10 \leqslant 70 - \sum\limits_{i=1}^{4} x_i + \xi_1 \leqslant 120,\\ 20 \leqslant 80 + x_2 - x_5 - x_6 + \xi_2 \leqslant 100,\\ 10 \leqslant 60 + x_3 - x_7 - x_8 + \xi_3 \leqslant 80,\\ 0 \leqslant 50 + x_4 + x_6 + x_8 - x_9 + \xi_4 \leqslant 90 \end{array}\right\} \geqslant 0.9,\\
&10 \leqslant x_1 \leqslant 50,\ 0 \leqslant x_2 \leqslant 10,\ 0 \leqslant x_3 \leqslant 10,
\end{aligned}$$

$$0 \leqslant x_4 \leqslant 15,\ 15 \leqslant x_5 \leqslant 60,\ -5 \leqslant x_6 \leqslant 5,$$
$$15 \leqslant x_7 \leqslant 60,\ \ -5 \leqslant x_8 \leqslant 5,\ \ 20 \leqslant x_9 \leqslant 70,$$
$$\xi_1 \sim \log N(2.24, 1.12),\ \ \xi_2 \sim \log N(1.60, 1.28),$$
$$\xi_3 \sim \log N(1.87, 1.45),\ \ \xi_4 \sim \log N(1.30, 1.34),$$
$$\eta \sim N(0, 2).$$

This is a nine-dimensional nonlinear JCCP problem with significance level 10%. The main difficulty in solving it is that the optimized quality is influenced seriously by four logarithmically normally distributed random variables of ξ_1, ξ_2, ξ_3 and ξ_4. Hence, solving such problem becomes difficult. After solving such problem, the statistical results obtained by the five approaches can be found in Table 2 (CI: confidence interval, AV: average of constraint violations, and AR: average runtime), and the average searching curves acquired by them are drawn in Fig. 1 (a). Note that the following figures are plotted through logarithm coordinates, because different algorithms have different maximal iterative numbers under the same maximal evaluation number (1.5×10^5). Through Fig. 1 (a) and the statistical values in Table 2, we can draw the following conclusions.

A) Optimized quality. The column on AV listed in Table 2 for this example shows that IOM and ASIA can obtain their respective reliable solutions during each single execution; other three approaches can only find some unreliable solutions after some executions. Therefore, the three approaches cannot compare with IOM and ASIA from the angle of solution quality. Through the values on mean, best, worst, standard deviation and AV, we can make a conclusion that ASIA can acquire the best average effect, while presenting stable average searching performance because of the small variance 3.307; IOM is secondary, even if it gets the maximal average value (147.552). SSGA-I and SSGA-II have similar effects, and HPSO works globally well over such two approaches.

Table 2 Comparison of statistical results for Example 1.

Algorithm	Best	Worst	Mean	Standard deviation	CI	AV	AR/s
HPSO	64.085	109.347	73.883	10.476	[71.431, 76.335]	0.047	16.32
SSGA-I	61.592	73.436	64.6927	2.734	[64.053, 65.333]	0.053	53.271
SSGA-II	61.898	67.721	64.420	1.659	[64.032, 64.808]	0.053	53.129
IOM	129.973	180.823	147.552	12.268	[144.680, 150.423]	0	26.199
ASIA	108.948	123.329	118.511	3.307	[117.737, 119.286]	0	17.692

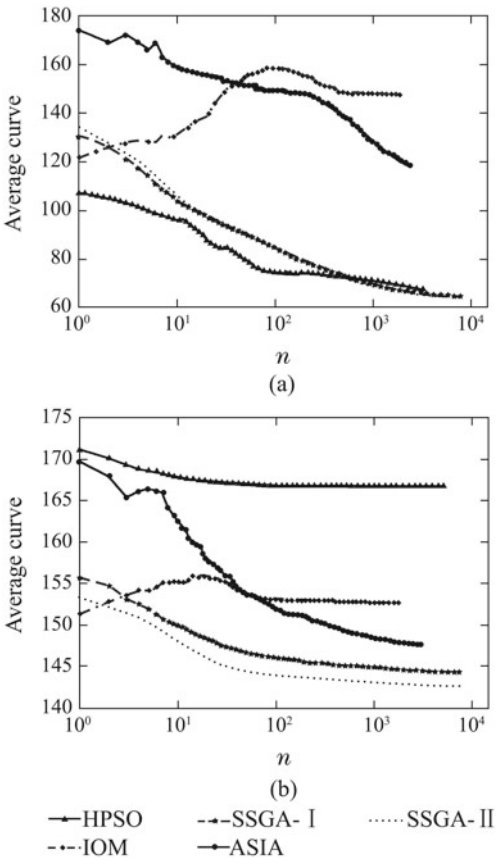


Fig. 1 Comparison on average searching curves for Examples 1 and 2. (a) Example 1. (b) Example 2.

HPSO only causes a relatively small average constraint violation, but it results in instable searching performance, due to a large variance (10.476). This hints that the training sample size directly influences the prediction ability of the BP network. Although SSGA-I and SSGA-II can only gain their inferior solution qualities, they have stable searching characteristics, because of small variances. This illustrates that a fixed sample size strategy helps SSGA-I and SSGA-II achieve stable searching behaviors. We also see that, despite of strong noises as in the above problem, ASIA can effectively urge the scheme of adaptive sampling to differentiate reliable and unreliable solutions, and thus enhance the ability of noisy suppression. In addition, the column on CI illustrates that the values on Mean acquired by the five algorithms belong to their respective confidence intervals with the same confidence level 90%. These intervals reveal some attributes of such approaches; namely, ASIA can get a narrow confidence interval, and thus converge to a desired approximate optimal reliable solution (see Fig. 1 (a)). Notice that, through such figure, it seems that HPSO, SSGA-I and SSGA-II have better searching performances than IOM and ASIA, because their searching curves are under those obtained by the latter two approaches when the iterative number becomes large. In fact, the former three approaches get early into local search, and can only find unreliable solutions.

B) Execution efficiency. Through the values on AR listed on the right side in Table 2, we know that HPSO spends a bit less time to solve the above problem than ASIA, but the average runtime demanded by the latter one is less than that spent by each of the other three approaches; namely, the average runtime of ASIA is at most 41% of that consumed by each of SSGA-I and SSGA-II, and 75% of that required by IOM.

Therefore, ASIA is effective and efficient for the above example; IOM is secondary, even if it has a lower efficiency than HPSO. Furthermore, we observe that HPSO is a static sampling approach, as it requires that the BP neural network be trained with 3000 samples before executing the optimization loop. Although it is efficient by comparison with SSGA-I, SSGA-II and IOM, it appears instable searching performance; SSGA-I and SSGA-II as static sampling optimization approaches have stable searching performances, but can only obtain some worse effects than HPSO. The above experimental results show that ASIA and IOM are indeed superior to other three optimization approaches, and meanwhile adaptive sampling is a more useful tool than static sampling.

Example 2 (Oil production planning) [2]

$$\begin{aligned}&\min \mathrm{E}[\eta + 2x_1 + 3x_2]\\ &\text{s.t.}\\ &\Pr\left\{\begin{array}{l}(2+\xi_1)x_1 + 6x_2 \geqslant 180 + \xi_3,\\ 3x_1 + (3.4-\xi_2)x_2 \geqslant 162 + \xi_4\end{array}\right\} \geqslant 0.8,\\ &x_1 + x_2 \leqslant 100,\ \ x_1 \geqslant 0,\ \ x_2 \geqslant 0,\\ &\xi_1 \sim U(-0.8, 0.8),\ \ \xi_2 \sim \exp(0.4),\\ &\xi_3 \sim N(0, 12),\ \ \xi_4 \sim N(0, 9),\ \ \eta \sim N(0, 2).\end{aligned}$$

Although this is a low-dimensional linear JCCP problem, the random variables ξ_1 and ξ_2 make sure that it is difficult to transform this problem into an analytically deterministic optimization problem. Moreover, the noisy intensity imposes the difficulty of solving such problem. Like the above experiment in Example 1, we acquire the statistical results in Table 3 and the average searching curves in Fig. 1 (b).

The values on mean, best, worst, standard deviation and AV in Table 3 show that for Example 2, IOM can find reliable solutions during each execution, but appears instable searching performance. Although ASIA can only find approximate solutions with extremely small constraint violations, it presents satisfactory performance, due to a small variance value (0.566). Whereas HPSO only causes the small average constraint violation relative to those caused by SSGA-I and SSGA-II, its solution quality during each execution is extremely instable, due to its large average variance (12.763). Relatively, it is superior to either SSGA-I or SSGA-II. Furthermore, we also note that SSGA-II and ASIA can all gain the narrow confidence intervals, but they expose different properties; in other words, Fig. 1 (b) indicates that ASIA is globally convergent and SSGA-II is locally convergent. Summarily, IOM can get the best effect, but it is instable; ASIA has stable searching performance, but it can only find approximate solutions; HPSO is instable, but it only causes the smaller constraint violation than either SSGA-I or SSGA-II. We also note that, although SSGA-I and SSGA-II appear stable searching behaviors, they present large constraint violations.

Through the values on AR in Table 3, the average runtime consumed by ASIA is at most 41% of that required by SSGA-I and SSGA-II, and about 74% of that spent by each of IOM; meanwhile, HPSO and ASIA have similar performance efficiencies. Therefore, with the aspect of execution efficiency, we can make the same conclusion as that acquired in Example 1.

Table 3 Comparison of statistical results for Example 2.

Algorithm	Best	Worst	Mean	Standard deviation	CI	AV	AR/s
HPSO	149.036	192.043	166.933	12.763	[164.971, 168.895]	0.0014	13.264
SSGA-I	142.069	155.298	144.281	3.128	[143.800, 144.762]	0.0927	37.889
SSGA-II	141.832	143.488	142.596	0.441	[142.529, 142.664]	0.0949	37.865
IOM	147.482	161.437	152.98	3.802	[152.396, 153.565]	0	20.494
ASIA	145.986	148.632	147.616	0.566	[147.529, 147.703]	0.0005	15.251

Example 3 (Multimodal function optimization) [31]

$$\begin{aligned}&\max \mathrm{E}[\eta + \sum_{k=1}^{3} x_k \sin(k\pi x_k)]\\ &\text{s.t.}\\ &\Pr\left\{\begin{array}{l}\xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 - 10 \leqslant 0,\\ \varsigma_1 x_1^2 + \varsigma_2 x_2^2 + \varsigma_3 x_3^2 - 100 \leqslant 0,\end{array}\right\} \geqslant 0.7,\\ &\xi_1 \sim U(0.8, 1.2),\ \ \xi_2 \sim U(1, 1.3),\\ &\xi_3 \sim U(0.8, 1.0),\ \ \varsigma_1 \sim N(1, 0.5),\\ &\varsigma_2 \sim \exp(1.2),\ \ \varsigma_3 \sim \log N(0.8, 0.6),\ \ \eta \sim N(0, 2).\end{aligned}$$

This is a three-dimensional, multimodal nonlinear JCCP problem. The main difficulty in finding the desired optimal reliable solution is that such problem includes many random variables with different distributions. These random variables seriously influence the process of identification whether a candidate is reliable or not. Like the above experiments, after the above five algorithms are applied to such example with respectively 30 single runs, we acquire the statistical results given in Table 4 and the average searching curves drawn in Fig. 2.

We know that the conclusion on performance effect for IOM and ASIA is the same as that obtained in Example 1. In other words, such two approaches can all find reliable solutions in any single run, but their effects are different. Obviously, ASIA has the better effect than IOM. Surprisedly, SSGA-I and SSGA-II are superior to HPSO because of their small constraint violations and variances, which is different from the conclusions presented in the experiments as Examples 1 and 2. The values on Mean hint that HPSO gets easily into local search. Furthermore, the confidence intervals acquired by the five approaches, together with the values on Mean and AV, show clearly that ASIA can converge to the global optimal reliable solution but other approaches can only be locally convergent, which can also be illustrated by Fig. 2. Through the values on AR in Table 4, we know that ASIA and HPSO need almost the same runtime to solve the problem. They have the same higher performance efficiency than any of other approaches. the average runtime consumed by ASIA is at most 35% of that required by ei-

ther SSGA-I or SSGA-II and at most 85% of that spent by IOM.

Totally, when solving each of the above problems, ASIA can obtain the best effect with high efficiency; IOM can get the better effect than each of other approaches, but presents somewhat low efficiency. HPSO has the better effect than SSGA-I and SSGA-II, but appears instable performance. Whereas SSGA-I and SSGA-II can only find inferior solutions with low efficiencies for Examples 1 and 2, but they have stable searching performances. In addition, all the above experimental results also illustrate that adaptive sampling is available for CCP problems, because it can make an optimization loop find valuable candidate solutions as fast as possible.

Table 4 Comparison of statistical results for Example 3.

Algorithm	Best	Worst	Mean	Standard deviation	CI	AV	AR/s
HPSO	1.992	−0.787	0.124	0.473	[0.079, 0.169]	0.0112	15.967
SSGA-I	9.581	8.320	9.101	0.331	[9.069, 9.133]	0.0064	45.717
SSGA-II	9.575	8.396	9.062	0.276	[9.035, 9.089]	0.0045	45.781
IOM	9.245	6.957	8.456	0.627	[8.396, 8.516]	0	18.813
ASIA	8.919	7.499	8.598	0.338	[8.566, 8.631]	0	15.934

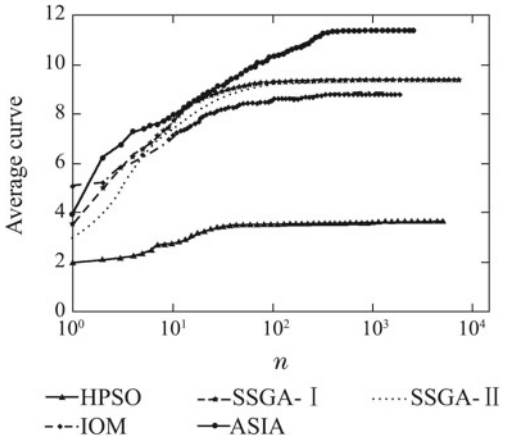


Fig. 2 Comparison on convergent curves for Example 3.

8 Conclusions

Stochastic optimization is an extremely challenging and active research topic in the field of optimization. We in this work make efforts to present a bio-inspired adaptive sampling immune algorithm for linear or nonlinear JCCP with a general distribution of the random vector. The focus for such approach is concentrated on how to estimate the joint chance constraint and to develop one scheme of sample allocation as well as immune evolution modules, and meanwhile we also concentrate on studying ASIA's computational complexity and convergence. Based on the central limit theorem, we acquire an *a priori* lower bound estimate for the probability displayed in the joint chance constraint. One such estimation, together with the concept of reliability-dominance, is utilized to help the proposed scheme of adaptive sampling decide sample sizes of different antibodies. These assist the immune modules to create valuable antibodies. Despite of strong noises presented in JCCP problems, such approach can effectively differentiate superior and inferior antibodies, and acquire the high-quality solution. It also reveals some excellent properties such as strong noise suppression, rapid evolution and convergence. The theoretical results demonstrate that ASIA is globally convergent. Our fundamental experiments, by comparison with those acquired by the four compared approaches, show that the proposed algorithm is an alternative JCCP optimizer which performs globally well over other approaches compared.

References

[1] W. van Ackooij, R. Zorgati, R. Henrion, et al. Chance constrained programming and its applications to energy management. *Stochastic Optimization – Seeing the Optimal for the Uncertain*. I. Dritsas (Ed.). Rijeka: InTech, 2011: 291 – 320.

[2] B. Liu, R. Zhao. *Stochastic Programming and Fuzz Programming*. Beijing: Tsinghua University Press, 1998.

[3] S. Ahmed, A. Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. *INFORMS: Tutorials in Operations Research*. Washington, 2008: http://www2.isye.gatech.edu/šahmed/cctutorial.pdf.

[4] A. Nemirovski, A. Shapiro. Convex approximations of chance constrained programs. *Society for Industrial and Applied Mathematics*, 2006, 17(4): 969 – 996.

[5] L. Hong, Y. Yang, L. Zhang. Sequential convex approximations to joint chance constrained programs: A Monte Carlo approach. *Operations Research*, 2011, 59(3): 617 – 630.

[6] R. Henrion, C. Strugarek. Convexity of chance constraints with independent random variables. *Computational Optimization and application*, 2008, 41(2): 263 – 276.

[7] S. Ahmed. Convex relaxations of chance constrained optimization problems. *Optimization Letters*, 2013: DOI 10.1007/s11590-013-0624-7.

[8] D. Dasgupta, S. Yu, F. Nino. Recent advances in artificial immune systems: Models and applications. *Applied Soft Computing*, 2011, 11(2): 1574 – 1587.

[9] A. Simões, E. Costa. CHC-based algorithms for the dynamic traveling salesman problem. *Applications of Evolutionary Computation*. Heidelberg: Springer-Verlag, 2011: 354 – 363.

[10] K. Trojanowski, S. T. Wierzchoń. Immune-based algorithms for dynamic optimization. *Information Sciences*, 2009, 179(10): 1495 – 1515.

[11] L. N. de Castro, J. Timmis. Artificial Immune Systems. *A New Computational Iintelligence Approach*. London: Springer-Verlag, 2002.

[12] V. S. Aragó, S. C. Esquivel. Optimizing constrained problems through a T-Cell. *Journal of Computer Science and Technology*, 2008, 8(3): 158 – 165.

[13] A. Shapiro. Monte Carlo simulation approach to stochastic programming. *Proceedings of the Winter Simulation Simulation Conference*. Arlington: IEEE, 2001: 428 – 431.

[14] J. Luedtke, S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 2008, 19(2): 674 – 699.

[15] B. K. Pagnoncelli, S. Ahmed, A. Shapiro. Sample average approximation method for chance constrained programming: Theory and applications. *Journal of Optimization Theory and Applications*, 2009, 142(2): 399 – 416.

[16] B. Liu, R. Zhao, G. Wang. *Uncertainty Programming and Its Applications*. Beijing: Tsinghua University Press, 2003.

[17] X. Ning. Solving stochastic chance-constrained programming problems with hybrid intelligent algorithm. *Computer Engineering and Applications*, 2010, 46(22): 43 – 46.

[18] C. A. Poojari, B. Varghese. Genetic algorithm based technique for solving chance constrained problems. *European Journal of Operational Research*, 2008, 185(3): 1128 – 1154.

[19] M. Brand. Reformulation of general chance constrained problems using the penalty functions. Stochastic Programming E-print Series (SPEPS), 2010: http://edoc.hu-berlin.de/series/speps/2010-2/PDF/2.pdf.

[20] J. L. Higle, L. Zhao. *Adaptive and Nonadaptive Samples in Solving Stochastic Linear Programs: A Computational Investigation*. Tucson: The University of Arizona, 2004.

[21] D. H. Loughlin, S. R. Ranjithan. Chance-constrained genetic algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco: Morgan Kaufmann, 1999: 369 – 376.

[22] C. Chen, J. Lin. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems: Theory and Applications*, 2000, 10(3): 251 – 270.

[23] C. Chen, D. He. Efficient simulation budget allocation for selecting an optimal subset. *Journal on Computing*, 2008, 20(4): 579 – 595.

[24] K. H. Sahin, U. M. Diwekar. Better optimization of nonlinear uncertain systems (BONUS): A new algorithm for stochastic programming using reweighting through kernel density estimation. *Annals of Operations Research*, 2004, 132(1/4): 47 – 68.

[25] Z. Zhang. Noisy immune optimization for chance-constrained programming problems. *Applied Mechanics and Materials*, 2011, 48(49): 740 – 744.

[26] Z. Zhang, X. Tu. Immune algorithm with adaptive sampling in noisy environments and its application to stochastic optimization problems. *IEEE Computational Intelligence Magazine*, 2007, 2(4): 29 – 40.

[27] M. W. Tanner. *New Solution Methods for Joint Chance-constrained Stochastic Programs with Random Left-hand Side*. Texas: Texas A & M University, 2009.

[28] M. W. Tanner, L. Ntaimo. IIS branch-and-cut for joint chance-constrained programs with random technology matrices. *European Journal of Operational Research*, 2010, 207(1): 290 – 296.

[29] U. Özcan. Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research*, 2010, 205(1): 81 – 97.

[30] Q. Bin. An immune algorithm for optional selection problem of investment projects. *The 5th International Conference on Natural Computation*. Tianjin: IEEE, 2009: 8 – 11.

[31] S. Mao, Y. Cheng, X. Pu. *Probabilistic Theory and Mathematical Statistic Tutorial*. Beijing: Higher Education Press, 2004.

[32] X. Huang, Z. Zhang, C. He, et al. *Modern Intelligent Algorithm: Theory and Applications*. Beijing: Science Press, 2005.

Appendix

Proof of Theorem 1 For a given antibody population A_n with size N and population sample size M_n, each antibody $\boldsymbol{x}$ in A_n needs to compute and compare $a_n(\boldsymbol{x})$ times through equations (1), (7) and (11), where $a_n(\boldsymbol{x}) = n(\boldsymbol{x})(I+2)+J+K+1$. Thus, Step 1 evaluate $Na_n(\boldsymbol{x})$ times. In Step 2, equation (9) indicates that A_n is required to execute antibody comparison with N^2 times and compute N times for deciding sample sizes of elements in A_n through equation (10), and hence, the total of computations and comparisons in Step 2 is as follows:

$$\begin{aligned} b_n &= \sum_{\boldsymbol{x}\in A_n} a_n(x) + N^2 + N \\ &= M_n(I+2) + N(J+K+2) + N^2. \end{aligned} \tag{a1}$$

In Step 3, dynamic proliferation needs at most $2N$ times to execute computation and population division through equations (12) and (13). In Step 4, we need at most c_n times to evaluate new antibodies and N times to calculate nutation probabilities, where $c_n = N(m_n(I+2)+J+K+1)$. Summarily, ASIA's computational complexity in the worst case is decided through

$$\begin{aligned} d_n =\ & Na_n(\boldsymbol{x}) + b_n + c_n + N = N[(m_0+m_n)(I+2) \\ & +3(J+K)+4] + M_n(I+2) + N^2. \end{aligned} \tag{a2}$$

Since $m_n \ll M_n$ in practical applications, it follows from equations (8) and (17) that

$$\begin{aligned} d_n &\leqslant N[(m_0+m_n)(I+2)+3(J+K)+4] \\ &\quad +m_0N\sqrt{1+n}(I+2)+N^2 \\ &= \mathrm{O}(m_0NI\sqrt{n}+N(J+K)+N^2). \end{aligned} \tag{a3}$$

Proof of Lemma 2 Write $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_p)$, $X = (X_1, X_2, \ldots, X_N)$ and $Y = (Y_1, Y_2, \ldots, Y_N)$. Through the definition of the polynomial mutation, if x_i is mutated into y_i, we acquire

$$y_i = x_i + d_i\Delta_i, \tag{a4}$$

$$d_i = \begin{cases} (2w)^{\frac{1}{1+\eta}} - 1, & \text{if } w \leqslant 0.5, \\ 1 - [2(1-w)]^{\frac{1}{1+\eta}}, & \text{if } w > 0.5, \end{cases} \tag{a5}$$

with $\eta > 0$ and $\Delta_i = [a_i \ \ b_i]$, where w is supposed to be a discrete random variable with uniform distribution on the set of equal division points in the interval $[0, 1]$. If $x_i \geqslant y_i$, it follows that

$$\begin{aligned} \Pr\{x_i \to y_i\} &= \frac{1}{2}\Pr\{w = \frac{1}{2}(1+\frac{y_i-x_i}{b_i-a_i})^{1+\eta}\} \\ &= \frac{1}{2m(a_i,x_i)} \geqslant \frac{1}{2r_i}, \end{aligned} \tag{a6}$$

where $\Pr\{x_i \to y_i\}$ and $m(a_i, x_i)$ are the transformation probability from x_i to y_i and the number of points between a_i and x_i in P_i, respectively. Similarly, if $x_i < y_i$, we can obtain

$$\begin{aligned} \Pr\{x_i \to y_i\} &= \frac{1}{2}\Pr\{w = 1 - \frac{1}{2}(1-\frac{y_i-x_i}{b_i-a_i})^{1+\eta}\} \\ &= \frac{1}{2m(x_i,b_i)} \geqslant \frac{1}{2r_i}. \end{aligned} \tag{a7}$$

Thereby, as associated to the design of the mutation rate of $p_m(\boldsymbol{x})$, the transformation probability $\Pr\{\boldsymbol{x} \to \boldsymbol{y}\}$ from $\boldsymbol{x}$ to $\boldsymbol{y}$ is given by

$$\begin{aligned} \Pr\{\boldsymbol{x} \to \boldsymbol{y}\} &= \prod_{i=1}^{p} \Pr\{x_i \to y_i\} \\ &\geqslant \delta_1 \equiv \frac{1}{\{\max\{2r_i\}\}^p} > 0. \end{aligned} \tag{a8}$$

Hence, for $X, Y \in \mathcal{S}^N$, it follows from equation (a8) that

$$\begin{aligned} \Pr\{X \to Y\} &= \prod_{i=1}^{N} p_m(X_i)\Pr\{X_i \to Y_i\} \\ &\geqslant \delta \equiv [\delta_1(1-\beta)]^N > 0. \end{aligned} \tag{a9}$$

In this way, the proof is completed.

Proof of Lemma 3 Through hypothesis H1) and Lemma 1, there exists $N_0 > 0$ such that when $n > N_0$, each element $\boldsymbol{x}^*$ in O_α^* is feasible in P_α^n. Therefore, for $n > N_0$, if $\boldsymbol{x}^* \in A_n$, $\boldsymbol{x}^*$ is a feasible candidate of P_α^n, and hence, we have that $\mathrm{aff}(\boldsymbol{x}^*) \geqslant \mathrm{aff}(\boldsymbol{x})$ with $\boldsymbol{x} \in A_n$ by the law of large number, that is, $\boldsymbol{x}^*$ is

best in A_n. Furthermore, if there exists some element $\boldsymbol{x}$ in C_n as in Step 4 of ASIA which is not inferior to $\boldsymbol{x}^*$, $\boldsymbol{x}$ must be a feasible candidate of P_α^n because $m_n > N_0$, and hence, $\boldsymbol{x}$ must be an optimal reliable solution to P_α. Summarily, when $n > N_0$, we have that $A_{n+1} \cap O_\alpha^* \neq \varnothing$ if $A_n \cap O_\alpha^* \neq \varnothing$. This shows that the conclusion is true.

Proof of Lemma 4 Let $A_n = X$ and $A_{n+1} = Y$ with $X \cap O_\alpha^* = \varnothing$ and $Y \subseteq O_\alpha^*$. Since the operation of proliferation in Step 3 is deterministic, there exists a unique state $Z_0 \in S^N$ such that $\Pr\{B_n = Z_0 | A_n = X\} = 1$. Hence, through K-C equation, it derives that

$$\begin{aligned}&\Pr\{A_{n+1} = Y | A_n = X\}\\ &= \sum_{Z \in S^N} \Pr\{C_n = Z | B_n = Z_0\} \Pr\{A_{n+1} = Y | C_n = Z\}.\end{aligned} \tag{a10}$$

Furthermore, through Step 5 and the proof of Lemma 3, there exists $N_0 > 0$ such that $\Pr\{A_{n+1} = Y | C_n = Y\} = 1$, and accordingly Lemma 2 and equation (a10) imply

$$\begin{aligned}&\Pr\{A_{n+1} = Y | A_n = X\}\\ &= \Pr\{C_n = Y | B_n = Z_0\} \geqslant \delta.\end{aligned} \tag{a11}$$

Consequently, we obtain

$$\begin{aligned}&\Pr\{A_{n+1} \cap O_\alpha^* \neq \varnothing | A_n \cap O_\alpha^* = \varnothing\}\\ &\geqslant \Pr\{A_{n+1} = Y | A_n = X\} \geqslant \delta.\end{aligned} \tag{a12}$$

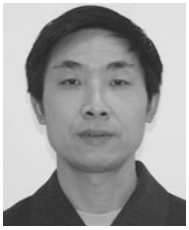
Proof of Theorem 2 For the same N_0 as in Lemma 3, Lemmas 2 and 3 imply

$$\begin{aligned}&\Pr\{A_{n+1} \cap O_\alpha^* = \varnothing\}\\ &= \Pr\{A_{n+1} \cap O_\alpha^* = \varnothing | A_n \cap O_\alpha^* = \varnothing\} \Pr\{A_n \cap O_\alpha^* = \varnothing\}\\ &\quad + \Pr\{A_{n+1} \cap O_\alpha^* = \varnothing | A_n \cap O_\alpha^* \neq \varnothing\} \Pr\{A_n \cap O_\alpha^* \neq \varnothing\}\\ &\leqslant (1 - \delta) \Pr\{A_n \cap O_\alpha^* = \varnothing\}.\end{aligned} \tag{a13}$$

Furthermore, by induction, it follows from equation (a13) that

$$\Pr\{A_n \cap O_\alpha^* = \varnothing\} \leqslant (1 - \delta)^{n - N_0}. \tag{a14}$$

Thus, the conclusion is true.



Zhuhong ZHANG received his M.S. degree from Department of Mathematics, Guizhou University, China, in 1998, and Ph.D. degree from College of Automation, Chongqing University, China, in 2004. Currently, he is a professor at the Guizhou University. His main areas of interests include uncertain programming, evolutionary algorithms, immune optimization, and signal simulation. E-mail: sci.zhzhang@gzu.edu.cn.

Lei WANG received his M.S. degree from Department of Mathematics, Guizhou University, China, in 2012. Currently, his main areas of interests include stochastic programming and immune optimization. E-mail: luckilywanglei@yahoo.cn.

Min LIAO received his M.S. degree from Department of Mathematics, Guizhou University, China, in 2012. Currently, his main areas of interests include dynamic optimization and immune optimization. E-mail: liaomin6984805@163.com.